计算机组成原理与系统结构

第9章 输入输出系统

杭州電子科技大學
HANGZHOU DIANZI UNIVERSITY

第9章 输入输出系统

BACK

9.1 概述

一 输入输出系统的构成

二 外设与CPU的连接

一、输入输出系统的构成

1. 外设的地位和作用
2. 外设的特点
3. 外设的分类
4. 外设的编址方式

1、外设的地位和作用

- 外部设备在计算机系统中的作用可以分为四个方面：

① 是人机对话的重要设备

交互

② 是完成数据媒体变换的设备

转换

③ 是计算机系统的软件和信息的驻在地

存储

④ 是计算机在各领域应用的重要工具

2、外设的特点

- 特点：
  - 工作速度差异大
  - 结构原理差异大
  - 时序独立、异步性明显
- 处理的信息不可能直接与CPU兼容：数据格式、逻辑时序不同
- 计算机与I/O设备间的连接与信息交换不能直接进行
- 必须设计一个“接口电路”作为两者之间的桥梁，使CPU和外设协调工作
- I/O接口电路：I/O适配器（I/O Adapter）

- 设置I/O接口的原因

① 外部设备工作的异步性：外部设备的工作时钟与时序是独立的，与微处理器的工作时序不同

② 速度上的差异：微处理器速度快，外部设备工作慢。

③ 信号线及数据格式不同

④ 有利于提高微处理器的工作效率：解放CPU

⑤ 便于外设自身的发展：多样化

3、外设的分类

❖ 磁带、磁盘、光盘（可读可写，既是输入设备又是输出设备）

❖ 按照功能：输入设备、输出设备

① 输入设备：各种形式的外部信息→计算机所能识别的二进制信息

- 键盘、鼠标、扫描仪、光笔、触摸屏、A/D转换器等

② 输出设备：计算机中的二进制信息→人或其他机器所能识别的信息形式

- 显示器、打印机、绘图仪、D/A转换器

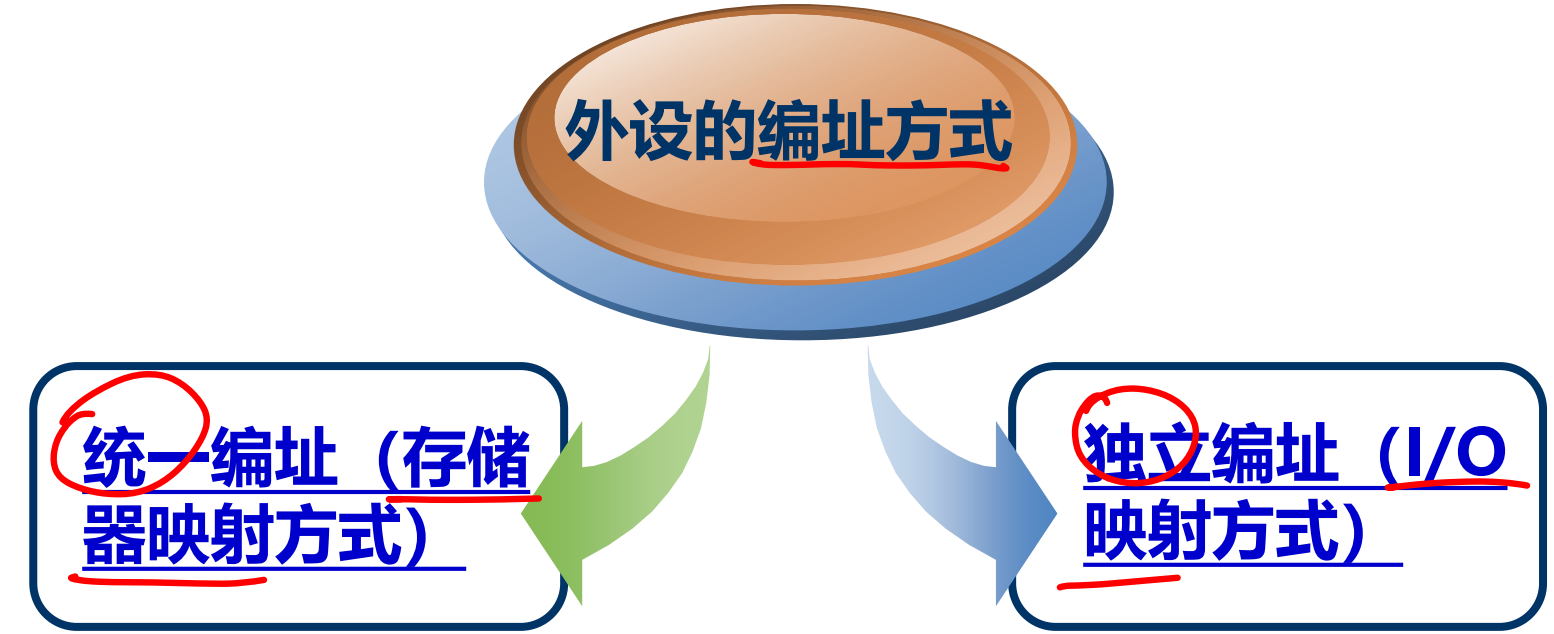
❖ 按照外设的工作速度：

- 低速设备（键盘、鼠标）、中速设备、高速设备（磁盘）

❖ 根据外设在计算机系统中所起的作用：

- 人—机交互设备、外存储器设备、通信设备

4、外设的编址方式



（1）统一编址

存储器映像

- 一个I/O端口等同于一个存储器单元
- 主存和I/O设备共用同一个地址空间
- 应用：Motorola 系列、Apple系列微型机及一些单片机和单板机
- 优点：

没有in/out指令

  - 指令系统：不设置专用的I/O指令，用功能很强的访存指令(如LOAD/STORE或者MOV)来访问I/O端口，通过地址来区分访问的是存储器还是I/O端口
  - 外设数目或I/O寄存器数几乎不受限制
  - 读写控制逻辑较为简单

可以没有IOR和IOW信号

（1）统一编址

- **缺点：**
  - I/O端口占用部分主存空间，可用主存空间减小；
  - 访存指令较长，执行速度较慢；
  - I/O端口地址译码电路复杂，译码时间较长。

- **举例：** CPU地址总线16位，前一半是主存地址空间，后一半是IO空间，硬件上如何实现？
  - A_{15}=0：主存的地址译码器工作，选中某个存储器单元读写
  - A_{15}=1：I/O设备的地址译码器工作，选中某个端口地址读写

（2）独立编址

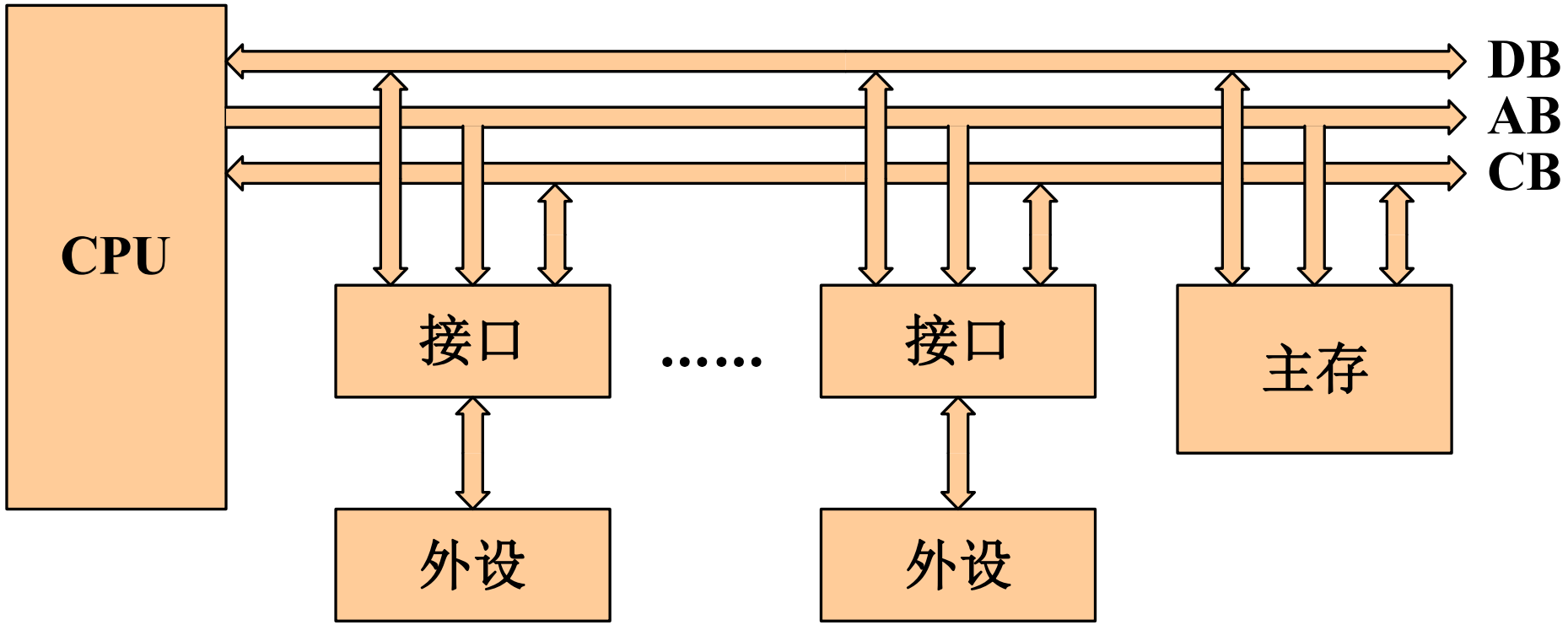
- I/O端口地址空间 与 存储器地址空间 相互独立。
- 指令系统：设置了专用的I/O指令，用I/O指令来访问I/O端口，用访存指令来访问存储器
- 虽然I/O端口地址与存储器地址有部分重叠，但通过指令可以区分。
- 应用：IBM-PC 系列、Z-80系列微型机及一些大型机。
- 优点：
  - I/O端口地址不占用存储器地址空间；
  - I/O端口数量不多，占用地址线少，地址译码简单，速度较快；
  - 使用专用I/O命令(IN/OUT)，指令短，执行速度快，可读性强。

（2）独立编址

- 缺点：
  - 专用I/O指令增加指令系统复杂性，且I/O指令类型少，程序设计灵活性较差
  - 要求处理器提供MEMR/MEMW和IOR/IOW两组控制信号，增加了控制逻辑的复杂性。
- 同一个系统中，可以同时使用两种方式。
- Intel系列微机I/O编址
  - Intel系列微处理器支持I/O独立编址方式，允许I/O统一编址（存储器映象编址）方式
  - Intel系列微机系统采用I/O独立编址方式

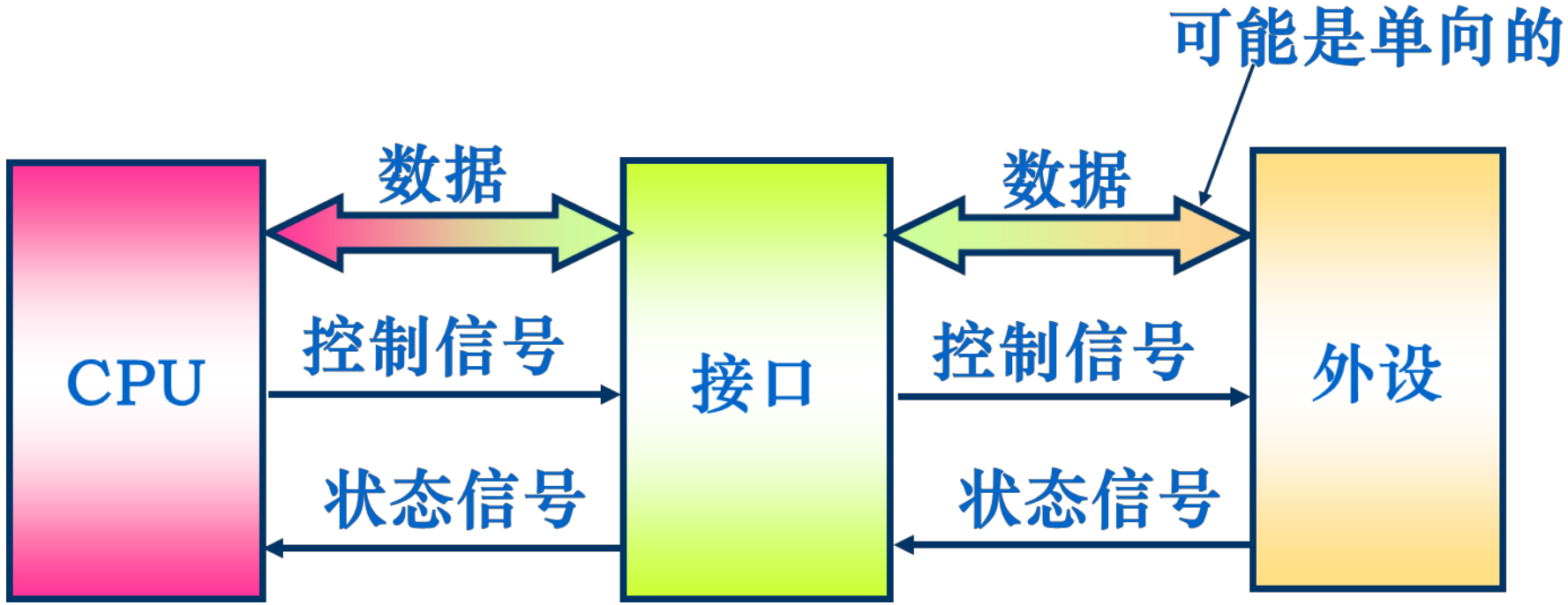
二、外设与CPU的连接

外设接口通过总线与CPU连接



❖ CPU访问外设的实质：访问外设接口中的寄存器(端口)。

CPU、I/O接口与I/O设备



- 相比存储器的访问，CPU访问外设的过程是完全等同的，不同的是所发送的读写信号有区别。

9.2 输入输出接口

一、I/O接口的功能

二、I/O接口的组成

一、I/O接口的功能

- I/O接口的功能如下：
  1. 实现数据缓冲
  2. 执行CPU的命令
  3. 返回外设的状态
  4. 设备选择
  5. 实现数据格式的转换
  6. 实现信号的转换
  7. 中断管理功能

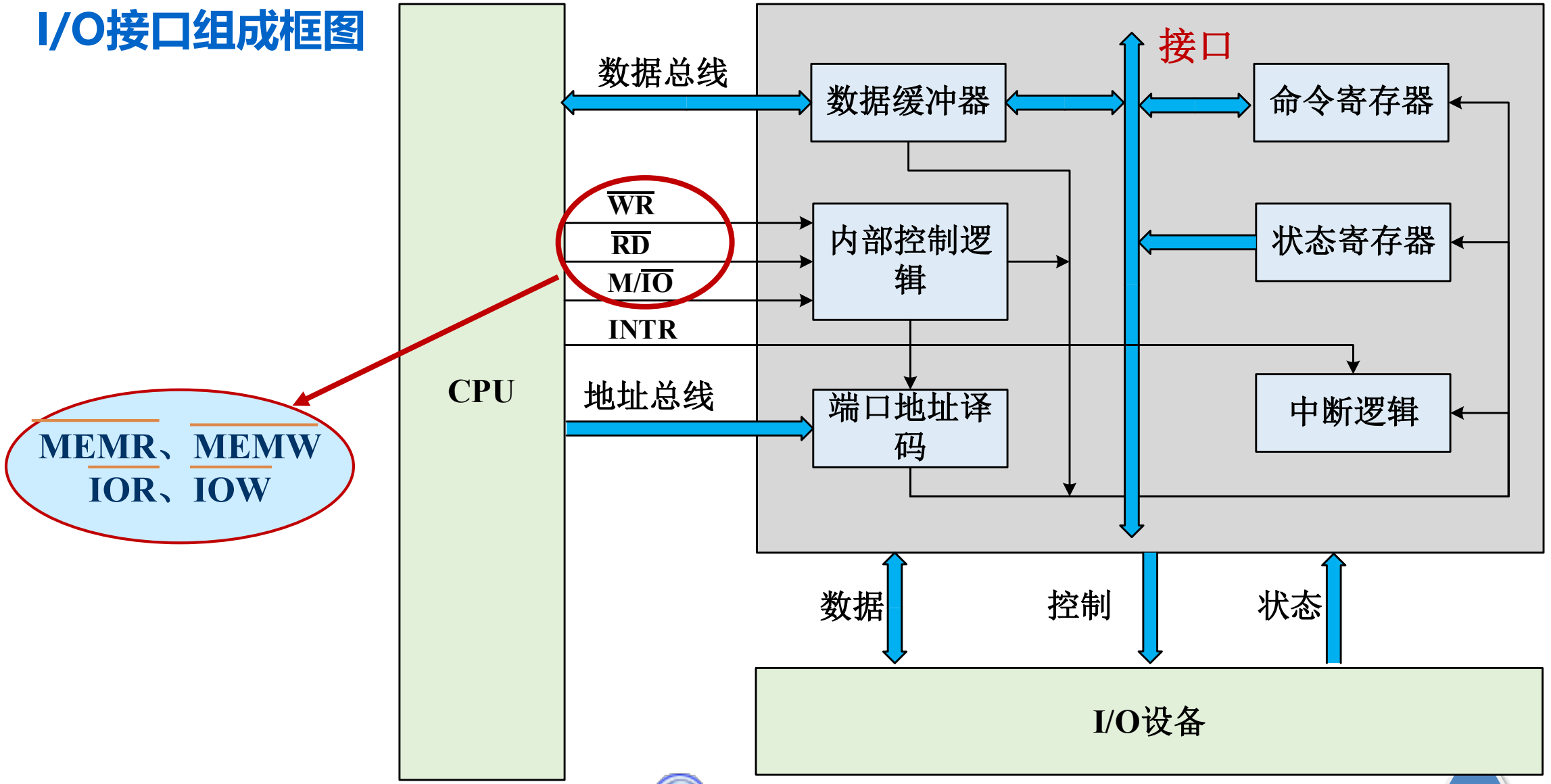
二、I/O接口的组成

❖I/O接口的硬件电路主要包括三部分：

- **基本电路：寄存器及其控制逻辑**
  - 命令寄存器（控制寄存器）及其译码器：保存CPU的命令
  - 数据缓冲寄存器：保存交换的数据
  - 状态寄存器：保存外设的状态
- **端口地址译码电路**：对地址总线上的外设地址进行译码，用以自我识别
- **供选电路**：中断控制逻辑、定时器、计数器、移位器等可选器件

二、I/O接口的组成

❖ I/O接口组成框图



9.3 主机与外设交换信息的方式

- 讨论的问题：CPU在何时、以什么方式对外设进行读写?
- 目标：传输效率高

4种方式名称
大致原理

一 程序查询方式

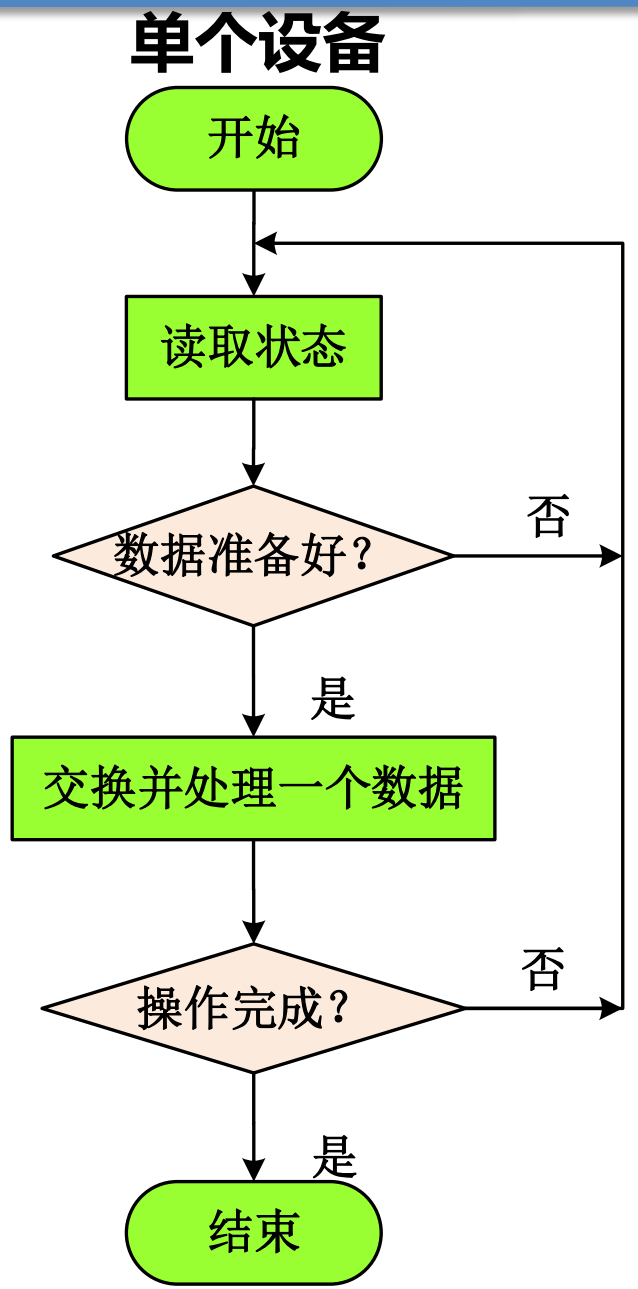
二 程序中断方式

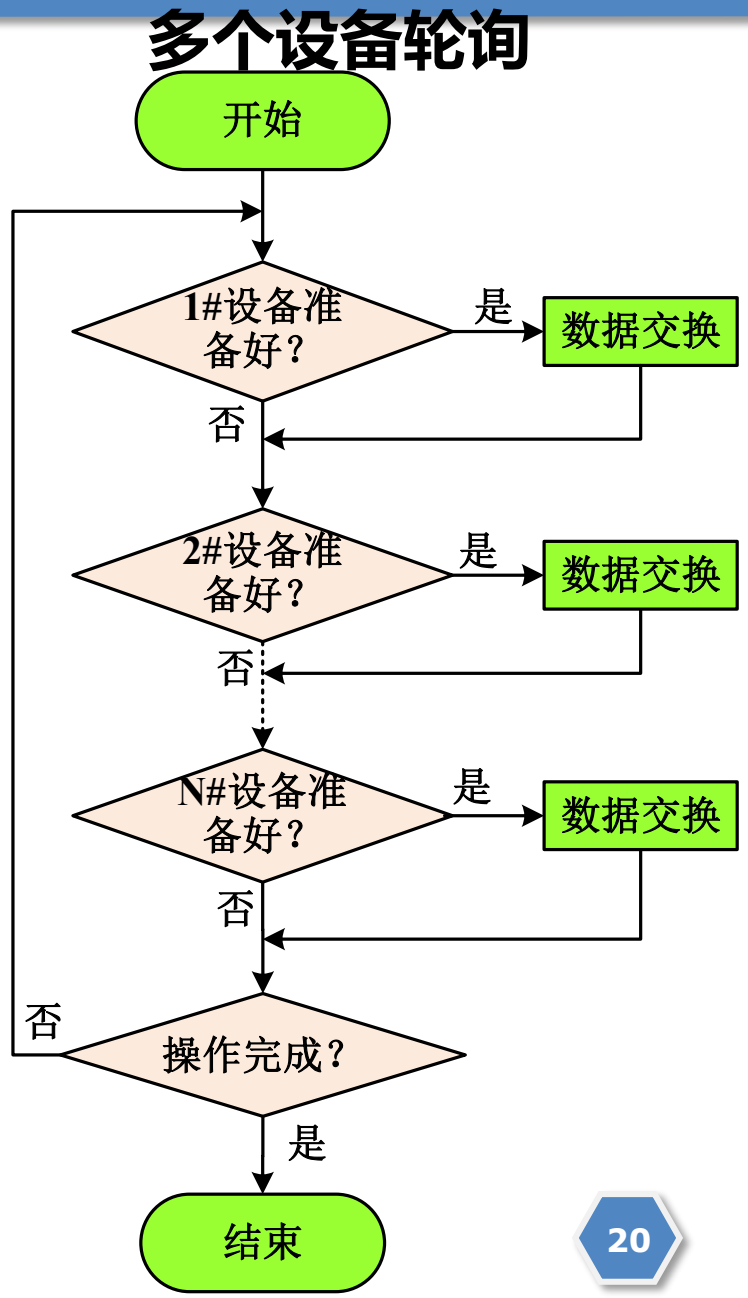
三 直接存储器访问（DMA）方式

四 通道与输入输出处理机方式

一、程序查询方式

- 工作原理：CPU查询外设已准备好后，才传送数据
- 特点：CPU与外设间通过程序同步，CPU被外设独占，CPU效率低下
- 要求：不需要增加额外的硬件电路
- 应用： 适用于在CPU不太忙且传送速度要求不高时



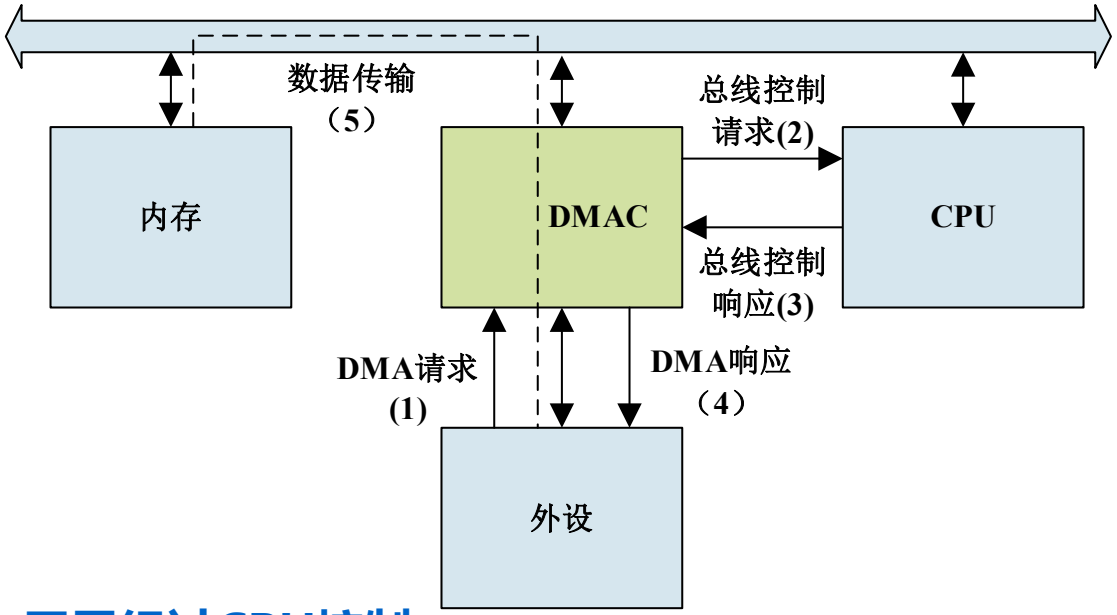


二、程序中断方式

- **工作原理：**
  - 外设准备数据时： CPU执行与传送数据无关的工作
  - 外设准备好数据后：主动向CPU发送一个中断请求
  - CPU执行完当前指令后：响应中断，自动转向中断服务程序
  - 在中断服务程序中：完成一个数据的传送
    - 执行IRET指令：中断返回至原来的断点处，继续执行
- **特点：在外设准备数据时，CPU与外设并行工作，CPU效率有所提高，并且CPU可以同时被多个外设占用**
- **要求：接口中需要中断控制逻辑支持**
- **应用：适用于中低速设备**

三、直接存储器访问（DMA）方式

- **工作原理：将I/O过程中，与内存交换数据的操作交由DMA控制器（DMAC）来控制，简化了CPU对输入输出的控制，进一步提高了CPU的效率**



- **①初始化阶段：CPU初始化DMA控制器，设置传送的方向、主存的起始地址和传送的数据字节数**
- **②数据传送阶段：**
  - （1）外设准备好数据，向DMA控制器发DMA请求信号
  - （2）DMA控制器向CPU申请占用总线
  - （3）若CPU响应了总线请求，则释放总线控制权
  - （4）DMAC控制器发出响应信号
  - （5）可以在主存和外设之间直接通过总线进行数据交换，无需经过CPU控制
  - 每传送一个数据，DMA控制器的字节计数器减1，如此重复传送，直至为0
- **③传送结束阶段：当CPU规定字节数的数据传送完毕后，DMA控制器通知CPU，CPU进行收尾，若出错，则需重传**

三、直接存储器访问（DMA）方式

- **特点：数据的传送不经过CPU（由DMAC控制），而I/O设备管理由CPU控制，简化了CPU对I/O的控制。硬件开销大，结构复杂，但CPU的效率高**
- **要求：需要DMA控制器及相关逻辑支持**
- **应用：适用于高速度大量数据传送时**

四、通道与输入输出处理机方式

- **通道**：是一个具有特殊功能的处理器，它可以实现对外围设备的统一管理和外围设备与内存之间的数据传送
- **特点**：能独立地执行用通道指令编写的输入输出控制程序，产生相应的控制信号送给由它管辖的设备控制器，继而完成复杂的输入输出过程
- **要求**：需要具有特殊功能的处理器，某些应用中称为输入输出处理器(IOP)
- **应用**：适用于高速度大量数据传送时
- **输入输出处理机(IOP)**：又称外围处理机(PPU)，它是通道方式的进一步发展。PPU基本上独立于主机工作，结构更接近一般处理机，甚至就是微小型计算机
- **特点**： IOP接管了CPU的各种I/O操作及I/O控制功能，CPU能与IOP并行工作
- **要求**：需要单独的处理机支持
- **应用**：高速I/O归IOP管理，低速I/O设备归CPU管理

9.4 中断系统

一、中断的基本概念

1. 中断源
2. 中断过程
3. 中断的作用

1、中断源

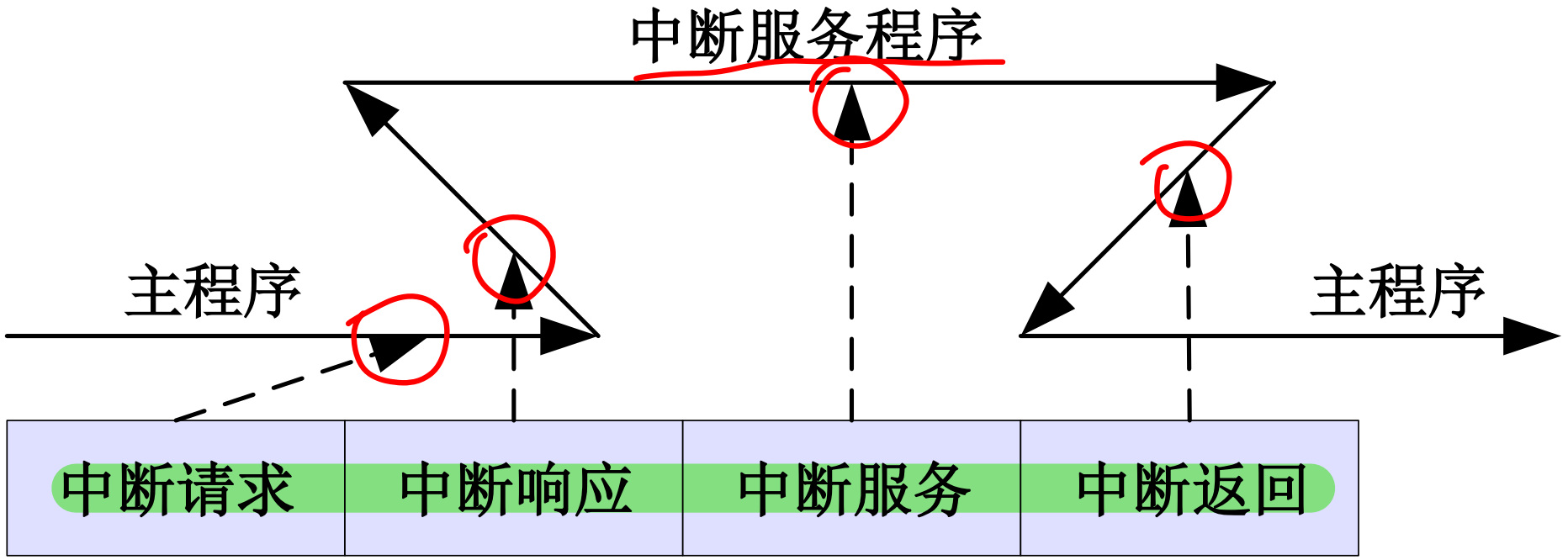
- ①中断：在CPU执行程序的过程中，由于某种事件发生，CPU暂时停止正在执行的程序而转向对所发生的事件进行处理，当对事件的处理结束后又能回到原来中止的地方，接着中止前的状态继续执行原来的程序，这一过程称为中断。
- ②中断源：由于某种原因引起CPU中断的事件或设备。
- 分为：硬中断、软中断两类。
- a.硬中断：由外部设备和其他CPU外部事件引起的中断，因此又叫外中断。
  - 常见外中断：输入输出请求、实时时钟、计时器、电源故障、设备故障、校验线路等等。
  - 外中断一般通过CPU的中断请求引脚引入。
    - 例如：在80X86系列CPU上，设有INTR、NMI两个中断请求引脚。

1、中断源

- b.软中断：指CPU内部的指令或程序执行中的突发事件所引起的中断，又叫内中断。
  - 常见软中断：指令中断（例如中断指令INT n）和程序异常（例如除数为零，运算溢出、指令的单步运行、程序运行至断点处等等）。
- ③中断类型号：对所有的中断源进行编码，为其分配的一个惟一的编号。
  - 例如，80X86有256种中断类型，因此中断类型号为8位二进制（0-255）。
- 中断类型号的作用：用于寻找中断服务程序的入口地址（中断向量），以实现程序转移。
- ④中断向量：指中断源对应的中断服务程序的入口地址。

2、中断过程

- 中断过程包含4个阶段



2、中断过程

① 中断请求（向CPU申请中断）

- 对于外中断，外设或其他中断源通过CPU的中断请求引脚向CPU发中断请求信号，CPU在每条指令执行完后，监测是否有中断请求，有则转入中断响应阶段。
- 对于内中断，则无需中断请求，直接可以根据中断类型号转入相应的中断服务程序。
- 需解决的主要问题是：
  a) 中断屏蔽：对那些CPU目前不准备响应的中断源，CPU如何禁止它们产生中断请求?
  b) 中断请求信号的传递：当系统中有多个中断源时，各中断源如何向CPU提出中断请求?
  c) CPU对中断请求信号的监测：CPU如何监测到有中断请求?

2、中断过程

②中断响应（转入中断服务程序）

- CPU执行中断隐指令（进入中断响应周期）
  - 通过硬件保存程序断点（PC）及标志寄存器→堆栈/特殊的寄存器
  - 通过向量方式/软件查询方式，得到中断服务程序入口，并置入PC
- 需解决的主要问题是：
  - 中断优先级的判别：如果同一时刻有多个中断源向CPU申请中断，CPU首先响应那个中断?
    - 中断排队与判优电路
  - 中断源的识别：CPU如何知道当前响应的是哪个中断源？即：转入哪个中断源的中断服务程序入口?
    - 向量方式/软件查询方式

2、中断过程

③中断服务（执行中断服务程序）

- 中断服务程序中
  - 1）保护现场，将有关寄存器的内容压栈
  - 2）处理中断，譬如进行I/O操作，实现数据传送
  - 3）恢复现场
- 需解决的主要问题是：
  - 中断嵌套：如果CPU在执行某个中断服务程序的过程中，又发生新的中断请求，那么CPU如何处理?

④中断返回（从中断服务程序返回）

- 执行IRET（中断返回指令），其功能是：将中断隐指令保存的程序断点和标志读出并送入PC和标志寄存器，从而回到CPU原来的程序断点处继续执行。

Interrupt Return

3、中断的作用

① 实现CPU和多台I/O设备并行工作

② 具有处理应急事件的能力

③ 进行实时处理

④ 实现人机通信

⑤ 实现多道程序运行和分时操作

⑥ 实现应用程序和操作系统的联系

⑦ 实现多机系统中各处理机间的联系

二、中断请求与判优

1. 中断请求信号的产生与监测
2. 中断屏蔽
3. 中断请求的排队判优

1、中断请求信号的产生与监测

- ①产生：

Interrupt Trigger

- 计算机的多个中断源随机向CPU发出中断请求，接口为每个中断源设置一个触发器，称为中断请求触发器INTR，当某个中断源有中断请求时，其相应的INTRi=1。
- 中断请求信号锁存在中断请求触发器中，等到CPU响应这个中断请求后才清除。
- 由多个中断请求触发器构成一个中断请求寄存器IRR，IRR每一位对应一种中断源。中断寄存器的内容称为中断字，中断字中为“1”的位表示对应的中断源存在中断。

IRR

$INTR_{n-1}$			$INTR_1$	$INTR_0$

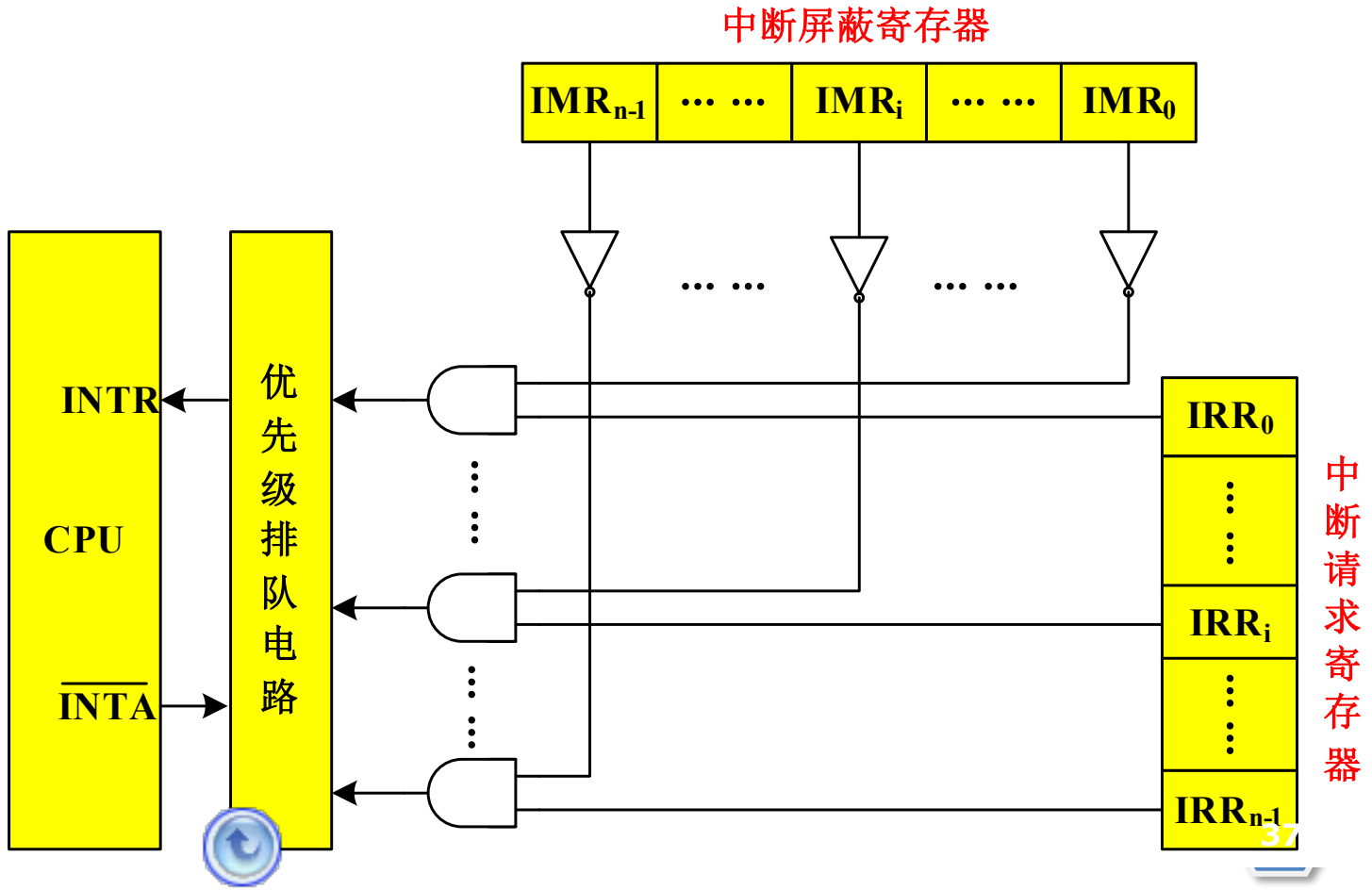
1、中断请求信号的产生与监测

- ②监测：
- CPU在每条指令执行完毕后，通过检测CPU的中断请求引脚是否有效来达到监测目的。
- CPU的中断请求引脚：多个，用以监测是否有中断发生。譬如，80X86CPU有INTR和NMI两条中断引脚。
  - INTR：可屏蔽的中断请求引脚，受程序状态字Flags的IF位（中断使能标志）的影响：IF=0，CPU禁止响应INTR引脚上的中断请求；IF=1，CPU允许响应INTR引脚上的中断请求。
  - NMI：不可屏蔽的中断请求引脚，不受IF的影响，一旦从该引脚引入的中断源有中断请求，CPU将会立即响应。

2、中断屏蔽

- 中断屏蔽触发器INTMi：每一个中断源单独设置一个
- 中断屏蔽寄存器IMR：将所有中断源的屏蔽触发器放在一起，构成寄存器，用一个地址对其寻址

  - INTMi=1，则中断源i被屏蔽(CPU不响应之)。
  - INTMi=0，则中断源i被开放(CPU响应之) 。



3、中断请求的排队判优

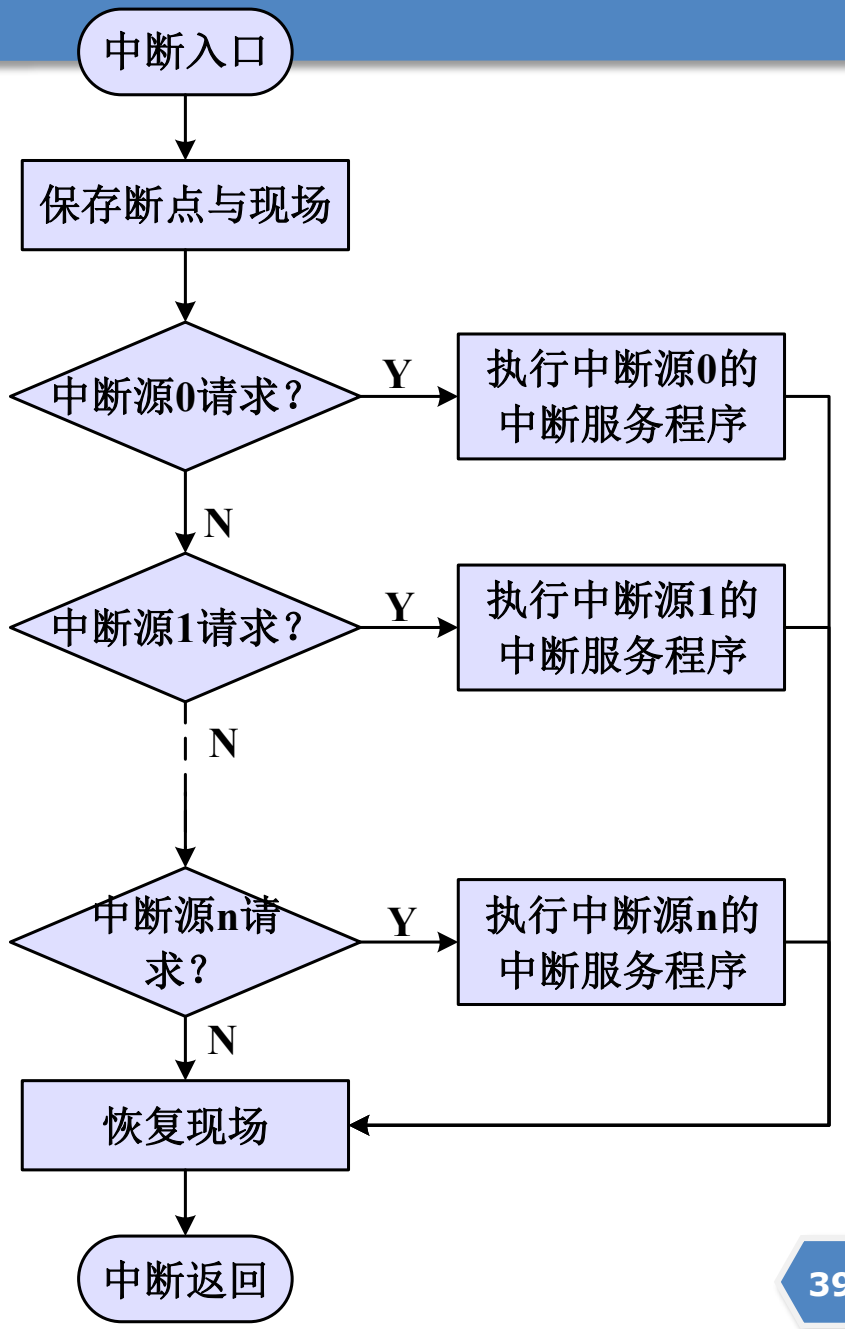
- **中断的优先级**：是指有多个中断同时发生时，CPU对中断源响应的次序。
- 确定中断优先级的原则是：
  - 对一旦提出请求需要立刻响应处理，否则就会造成严重后果的中断源，设定最高的优先级
  - 对可以延迟响应和处理的中断源，设定较低的优先级。
  - 一般，把硬件故障引起的中断优先级定为最高，其次是软件故障中断和I/O中断。

3、中断请求的排队判优

- 中断请求的排队判优，常用的方法有两种：软件查询和硬件排队电路

①软件查询法：用程序来判断优先级，这是最简单的中断判优方法。软件查询法用于一根公共请求线的情况

- 优点：可以灵活地修改中断源的优先级别，硬件电路实现简单；
- 缺点：查询、判优完全靠程序实现，需要占用CPU时间，同时中断响应较慢，优先级较低的设备被响应的等待时间也较长。



3、中断请求的排队判优

②硬件排队电路

- 优先级别高的中断请求将自动封锁优先级别低的中断请求的处理。硬件排队电路一旦设计连接好之后，将无法改变其优先级别。

比如，串行排队链

- 中断响应信号逐级传送，先到达的设备，其优先级高于中断响应信号后到达的设备，即电路中距离CPU最近的中断源优先级最高，这里距离远近是指电气上的信号传递顺序。这种方法实现时电路较简单，但优先级固定，取决于固定的硬件连接，不够灵活，不易于改变或调整优先级。

三、中断响应

1. CPU响应中断的条件

① CPU的中断使能触发器开放（IE=1，允许中断）；

② CPU的中断请求引脚有效；

③ 当前指令执行完；

2. CPU中断响应的过程：执行中断隐指令

① 关中断

- 当CPU响应中断后，立即自动关中断（把内部的中断使能触发器IE清零），禁止接收新的中断，以保证接下来中断隐指令操作不被打断。

② 保存断点

- 断点信息包括两部分：PC和程序状态字PSW
- 通常保存在堆栈中，有些计算机中将断点保存在特殊的中断返回寄存器中

三、中断响应

2. CPU中断响应的过程

③ 识别中断源，转入服务程序入口地址

- 中断源识别的方法有两种：向量中断和程序查询

a) 向量中断

- 中断向量：中断服务程序入口地址
- 中断向量表：各中断源的中断向量存放在内存一片连续的单元中，形成一张表
- 中断向量地址：中断向量在中断向量表中的单元地址
- 使用向量识别中断源：当CPU响应中断时，由硬件（外设接口或者中断控制器）自动产生一个指定的地址（向量地址）或者代码（中断类型号），它们与该中断源的中断向量有一一对应关系。由向量地址或中断类型号指出每个中断源设备的中断向量（中断服务程序入口地址）

三、中断响应

b) 程序查询：

- **轮询**：由CPU执行一个公共的中断处理程序，逐个询问外设接口有否发出中断请求（测试中断请求触发器），若有中断请求，则转入其中断服务程序的入口开始执行。

四、中断服务与返回

1. 中断服务程序
2. 中断服务程序与子程序的区别
3. 中断嵌套

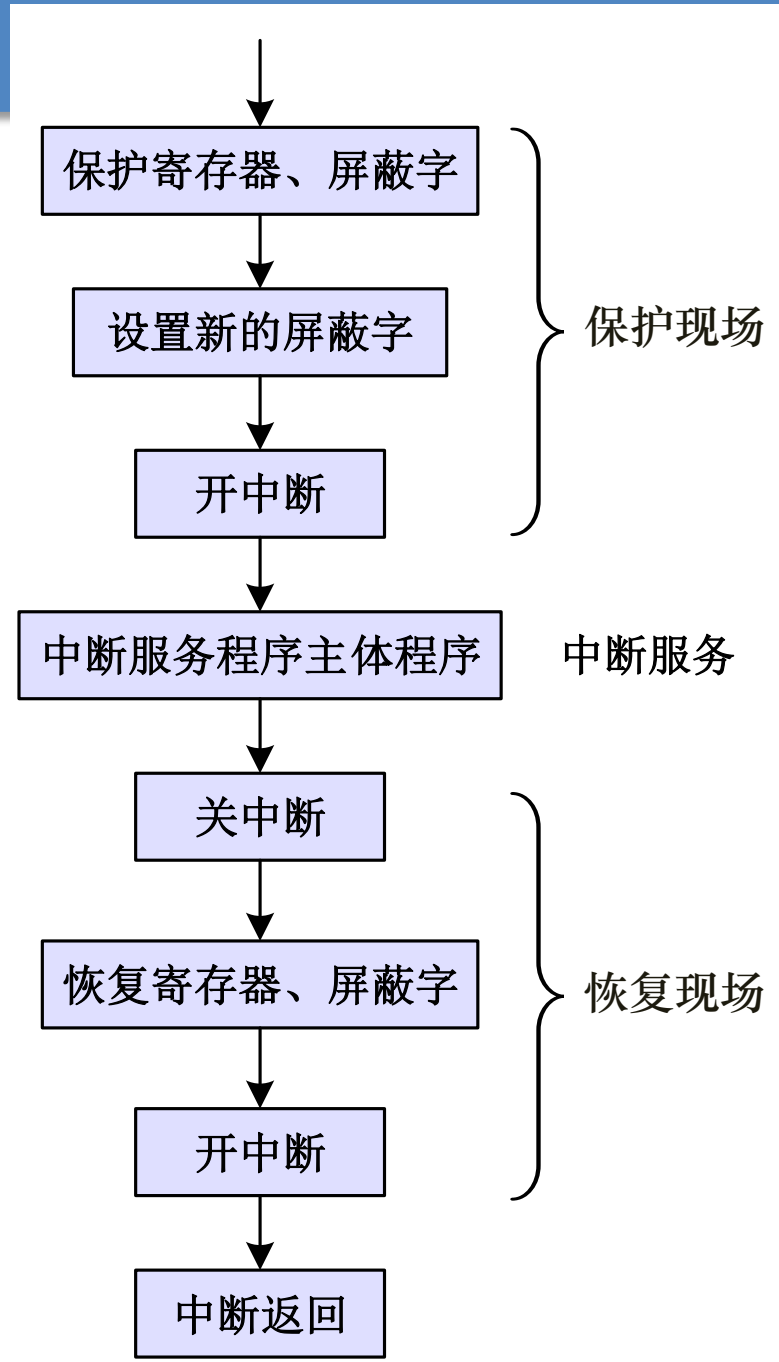
1、中断服务程序

① 保护现场

② 中断服务

③ 恢复现场

④ 中断返回



2、中断服务程序与子程序的区别

① 调用或者触发的主体不同：

- 子程序的执行：由程序员事先安排好的（调用子程序指令转入）；主动性
- 中断服务程序的执行：是由随机的中断事件引起的；被动性

② 与主程序的关系：

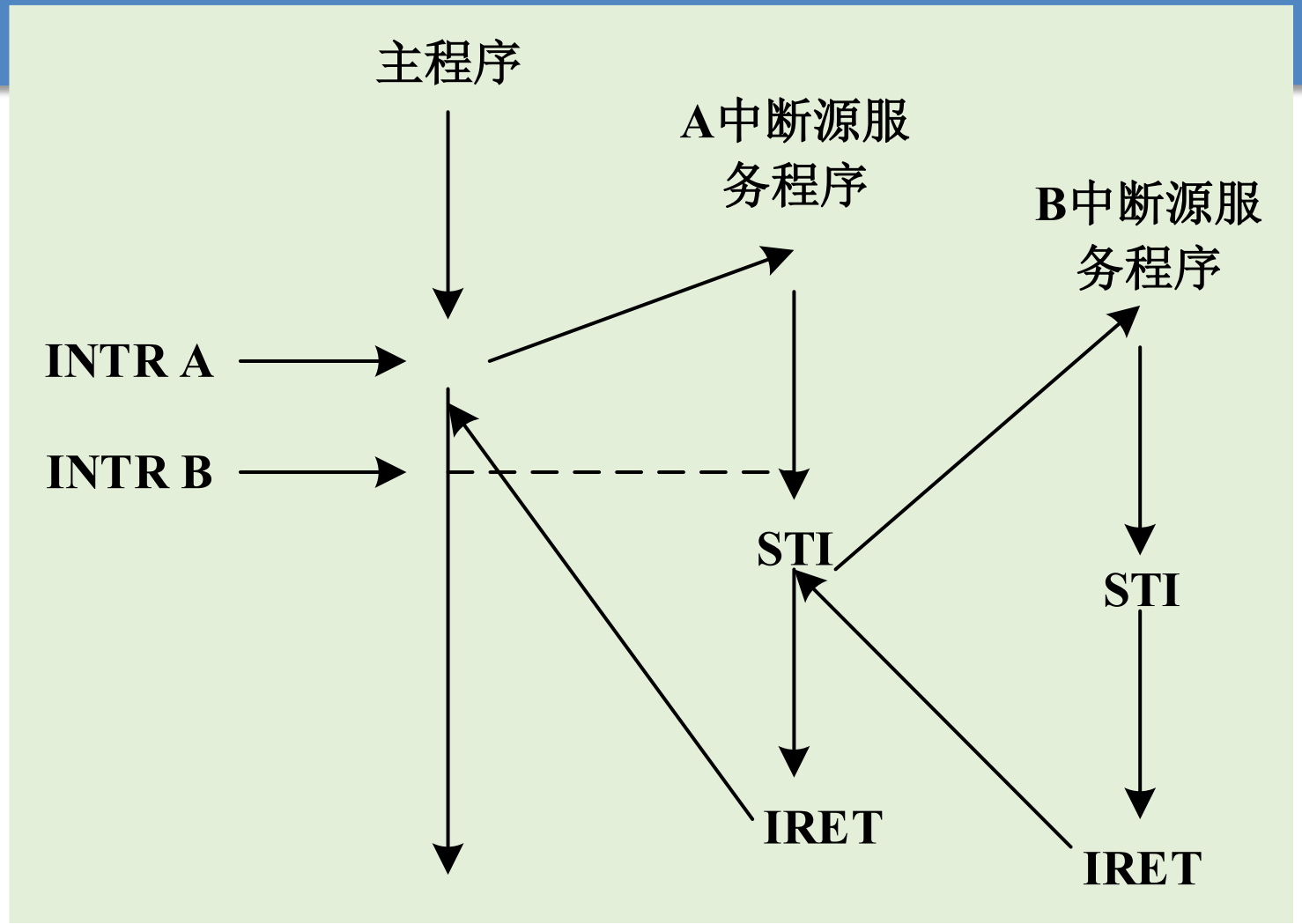
- 子程序：受到主程序或其上层程序的控制
- 中断服务程序：一般与被中断的现行程序毫无关系

③ 并发性：

- 不存在同时调用多个子程序的情况，而有可能发生多台I/O设备同时请求CPU为自己服务的情况。

3、中断嵌套

- **中断嵌套**：是指CPU在执行某个中断服务程序的过程中允许再响应更高级别的中断请求，也称为多重中断。
- **单重中断**：正在执行的中断服务程序中，禁止再响应其他中断请求



- 中断嵌套技术的实现，关键是在中断处理程序中必须适时开放中断（STI指令），并且使用堆栈的“先进后出”特性保证中断的逐级返回。

set interrupt

本章小结

- **外设的编址方式**：统一编址和独立编址
- CPU访问外设，就是CPU访问外设接口(中的寄存器)。
- 在计算机的输入输出系统中，**主机与外设交换信息的方式**有四种：
  - （1）程序查询方式；
  - （2）程序中断方式；
  - （3）直接存储器访问（DMA）方式；
  - （4）通道与输入输出处理机（IOP）方式。
- **中断技术**实现了CPU与外设之间并行工作，提高了输入输出效率，同时它不仅是计算机处理一切随机出现的事件的手段，而且也是实现计算机系统资源管理的重要方法。

本章小结

- **中断过程**：包含中断请求、中断响应、中断服务、中断返回4个阶段。
- **中断屏蔽技术**：不仅使得CPU能够禁止或允许某些中断源的中断请求，并且可以灵活地修改中断源的优先级。
- **中断请求优先级的排队方法**：软件查询和硬件排队判优两种方法。
- **排队后的中断请求信号**：可以单线、多线或者二维结构的形式传送至CPU。
- CPU则**在本条指令执行完后**，监测中断输入引脚有否中断，若有中断请求，则通过**中断隐指令**来保存断点和识别中断源，并转入中断服务程序执行。
- 中断服务程序的最后，通常以**中断返回指令**，返回断点处。

The End !